ATLAS EXPERIMENT

OpInt activities from Rucio

D. Bavarajan, T.Javurek, M. Lassnig, C. Serfon

Introduction

- A lot of actions done by the shifters/DDM operations are repetitive tasks
- During the last years a lot of work has been done to automatise some of these tasks, e.g. :
  - Detection and recovery of lost files
  - Identification of Dark Data
  - Automatic rebalancing of data
- For all these tasks, specific services were developed. But :
  - Very specific (Cannot do more than what they are supposed to do)
  - Not experiment agnostic (all of them rely on Rucio)
  - Cannot evolve and get more performant with time

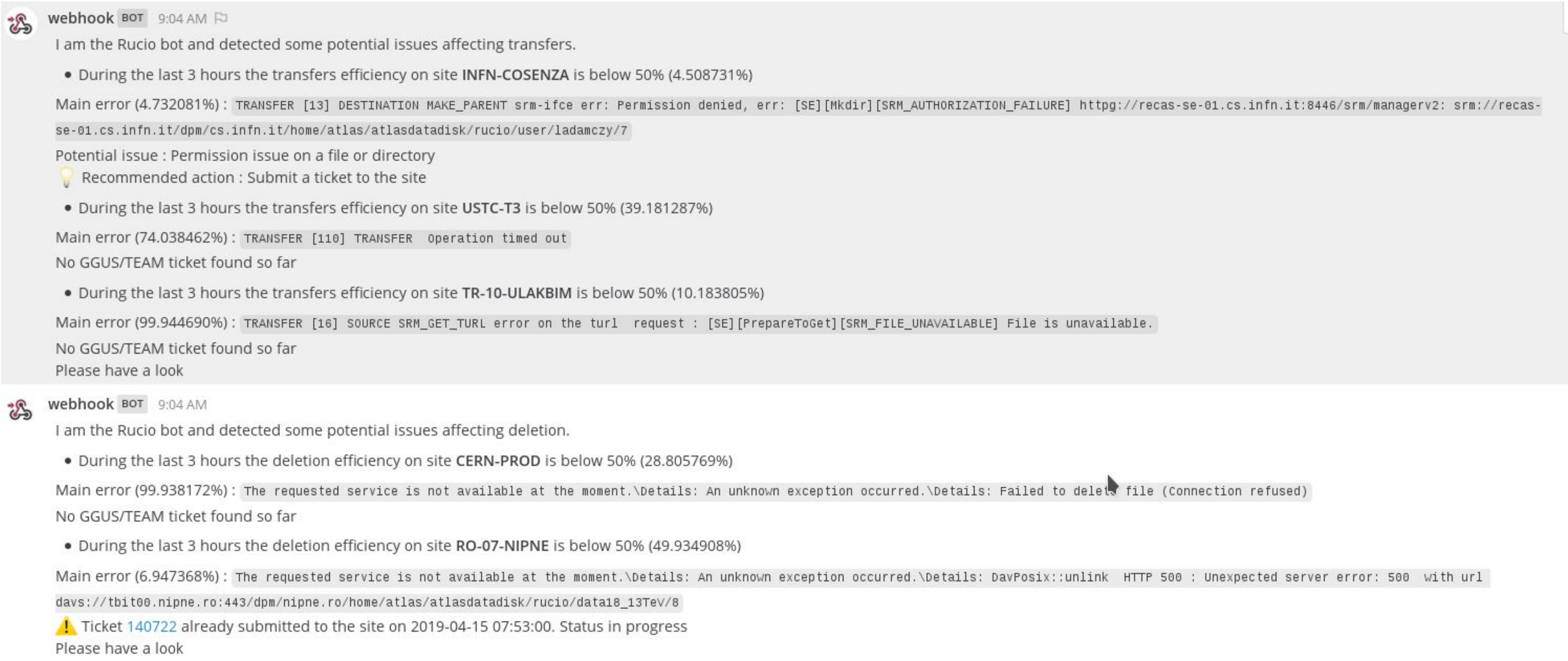
What do we want to do ?

- New generic tool (expert system) being developed :
  - Will identify recurrent errors in transfers, deletions
  - Suggest to the shifters the actions to take. The shifter will feed back to the service if the suggested action was appropriate, helping the system to become more efficient with time
  - In the medium term, for well identified errors, can even act for the shifters (e.g. submit tickets)
- Main idea is to start simple to address immediate needs from operation team
  - I.e. no fancy feature like ML for the time being
  - Should be operational quickly (not in 2 years)

Proof of concept

- Proof of concept done by writing small sets of scripts that collect the information from DDM ElasticSearch
  - A list of known errors patterns and associated issues is recorded into a knowledge DB (simple text file)
  - The issues are associated to actions suggested to the shifters
  - The scripts are run regularly and inform the shifters of most common errors and suggested action (send notification on Mattermost)
  - If GGUS tickets are already submitted for the faulty site, inform the shifters
- Missing functionalities :
  - No possibility to query the service/provide feedback

Proof of concept



webhook BOT 9:04 AM

I am the Rucio bot and detected some potential issues affecting transfers.

- During the last 3 hours the transfers efficiency on site **INFN-COSENZA** is below 50% (4.508731%)

Main error (4.732081%) : `TRANSFER [13] DESTINATION MAKE_PARENT srm-ifce err: Permission denied, err: [SE][Mkdir][SRM_AUTHORIZATION_FAILURE] httpg://recas-se-01.cs.infn.it:8446/srm/managerv2: srm://recas-se-01.cs.infn.it/dpm/cs.infn.it/home/atlas/atlasdatadisk/rucio/user/ladamczy/7`

Potential issue : Permission issue on a file or directory
💡 Recommended action : Submit a ticket to the site

- During the last 3 hours the transfers efficiency on site **USTC-T3** is below 50% (39.181287%)

Main error (74.038462%) : `TRANSFER [110] TRANSFER  Operation timed out`

No GGUS/TEAM ticket found so far

- During the last 3 hours the transfers efficiency on site **TR-10-ULAKBIM** is below 50% (10.183805%)

Main error (99.944690%) : `TRANSFER [16] SOURCE SRM_GET_TURL error on the turl  request : [SE][PrepareToGet][SRM_FILE_UNAVAILABLE] File is unavailable.`

No GGUS/TEAM ticket found so far
Please have a look

webhook BOT 9:04 AM

I am the Rucio bot and detected some potential issues affecting deletion.

- During the last 3 hours the deletion efficiency on site **CERN-PROD** is below 50% (28.805769%)

Main error (99.938172%) : `The requested service is not available at the moment.\Details: An unknown exception occurred.\Details: Failed to delete file (Connection refused)`

No GGUS/TEAM ticket found so far

- During the last 3 hours the deletion efficiency on site **RO-07-NIPNE** is below 50% (49.934908%)

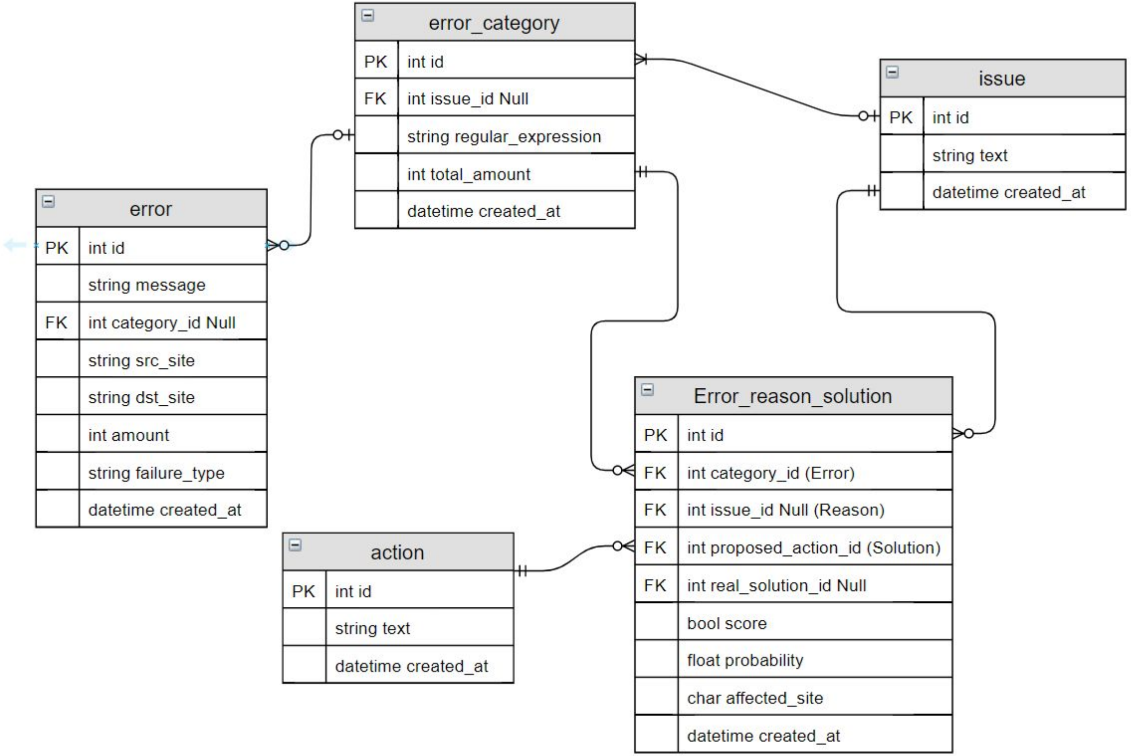
Main error (6.947368%) : `The requested service is not available at the moment.\Details: An unknown exception occurred.\Details: DavPosix::unlink  HTTP 500 : Unexpected server error: 500  with url davs://tbit00.nipne.ro:443/dpm/nipne.ro/home/atlas/atlasdatadisk/rucio/data18_13TeV/8`

⚠️ Ticket 140722 already submitted to the site on 2019-04-15 07:53:00. Status in progress
Please have a look

Implementation of the new service

- Work started (D. Bavarajan) to have a proper service based on a server/client infrastructure
- On the server side, a Database records all the errors collected from ElasticSearch
  - The errors are mapped to a list of defined generic errors (regular expression). The definition of the generic error is currently done manually, but evaluating if a Natural Language Processing can do the work
  - In a bootstrap phase we associate all the generic errors to a list of suggested action. The association is weighted by a probability that will be evolve with the shifter feedback
  - A cron job will collect regularly the new errors

DB schema



error_category

PK	int id
FK	int issue_id Null
	string regular_expression
	int total_amount
	datetime created_at

issue

PK	int id
	string text
	datetime created_at

error

PK	int id
	string message
FK	int category_id Null
	string src_site
	string dst_site
	int amount
	string failure_type
	datetime created_at

Error_reason_solution

PK	int id
FK	int category_id (Error)
FK	int issue_id Null (Reason)
FK	int proposed_action_id (Solution)
FK	int real_solution_id Null
	bool score
	float probability
	char affected_site
	datetime created_at

action

PK	int id
	string text
	datetime created_at

Interface with humans/other services

- A REST interface is being developed so that the shifters can interact with the service. It will provide functionality to :
  - Query for errors recently reported and the potential issue/action associated
  - Report what the issue actually was and what action was taken
  - Post new errors (to interface with other services)
- Implementation will be done using flask
- Interface with other services for the notification and identification of ticket already submitted will be developed :
  - Easy for services that provide REST interface, e.g. : Mattermost, JIRA, SLAC
  - More difficult for GGUS (SOAP)

Status

- Database design : Done. Use of SQLAlchemy to be independent of the backend
- Bootstrap of the list of generic errors : Ongoing
- Definition and implementation of the REST interface : Ongoing. Expected by the end of July
- Interface with other services (JIRA, Mattermost, GGUS). To be done
- Longer term :
    - Natural language processing to provide interface to the end-users (e.g. why my transfers are stuck). See next slide
    - Machine learning ?

Natural Language processing

- What is NLTK:
  - https://www.nltk.org/book/
- Although NLTL is designed for Natural Language, it can well analyze also structured text.
- Main features of NLTK:
  - Semantics, Context, e.g. can compare similarities in sentences
  - Frequency distributions, e.g. searching for frequent patterns
  - Sentiment (Positive, Negative, Neutral)
  - etc.
- Our use cases could be:
  - Popularity of datasets/scopes from emails sent to specific e-groups (next slide)
  - Classification of error messages

Rucio is watching you!!!

- Usage of NLTK to process emails going to e-groups
- Motivation
  - correlation of content of the emails to the data popularity
  - effective support in case of any complains
  - penalties in case of too many complains :-D
- Roughly 5000 emails/day is already good statistics

Semantics Test with NLTK:
Rucio makes me happy every day.
compound: 0.5719, neg: 0.0, neu: 0.575, pos: 0.425,
Rucio makes me happy.
compound: 0.5719, neg: 0.0, neu: 0.448, pos: 0.552,
Rucio makes Rod crying.
compound: -0.4767, neg: 0.508, neu: 0.492, pos: 0.0,

Conclusion

- A service is being developed to offload operation team/shifters of the most common errors linked to transfers/deletion
- Timescale to have a first usable prototype : end of summer
- The tool will be :
  - Experiment agnostic
  - Workflow agnostic (i.e. the errors collected can be Data Management related or Workflow Management related)
- The code will be committed into the Opint github repository
- People interested in the project can join